

академия
больших
данных

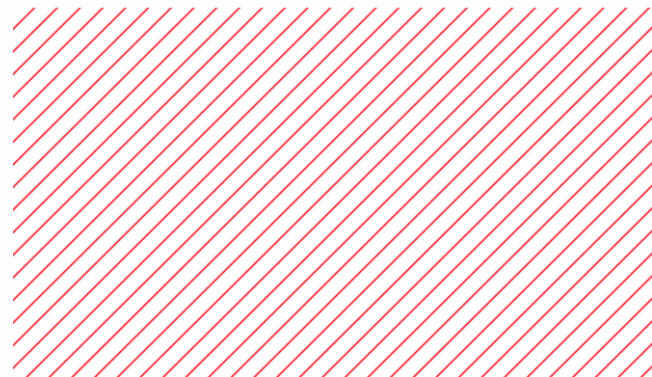
mail.ru
group

made

Фильтрация изображений и выделение краев

Андрей Савченко

Профессор НИУ ВШЭ-Нижний Новгород

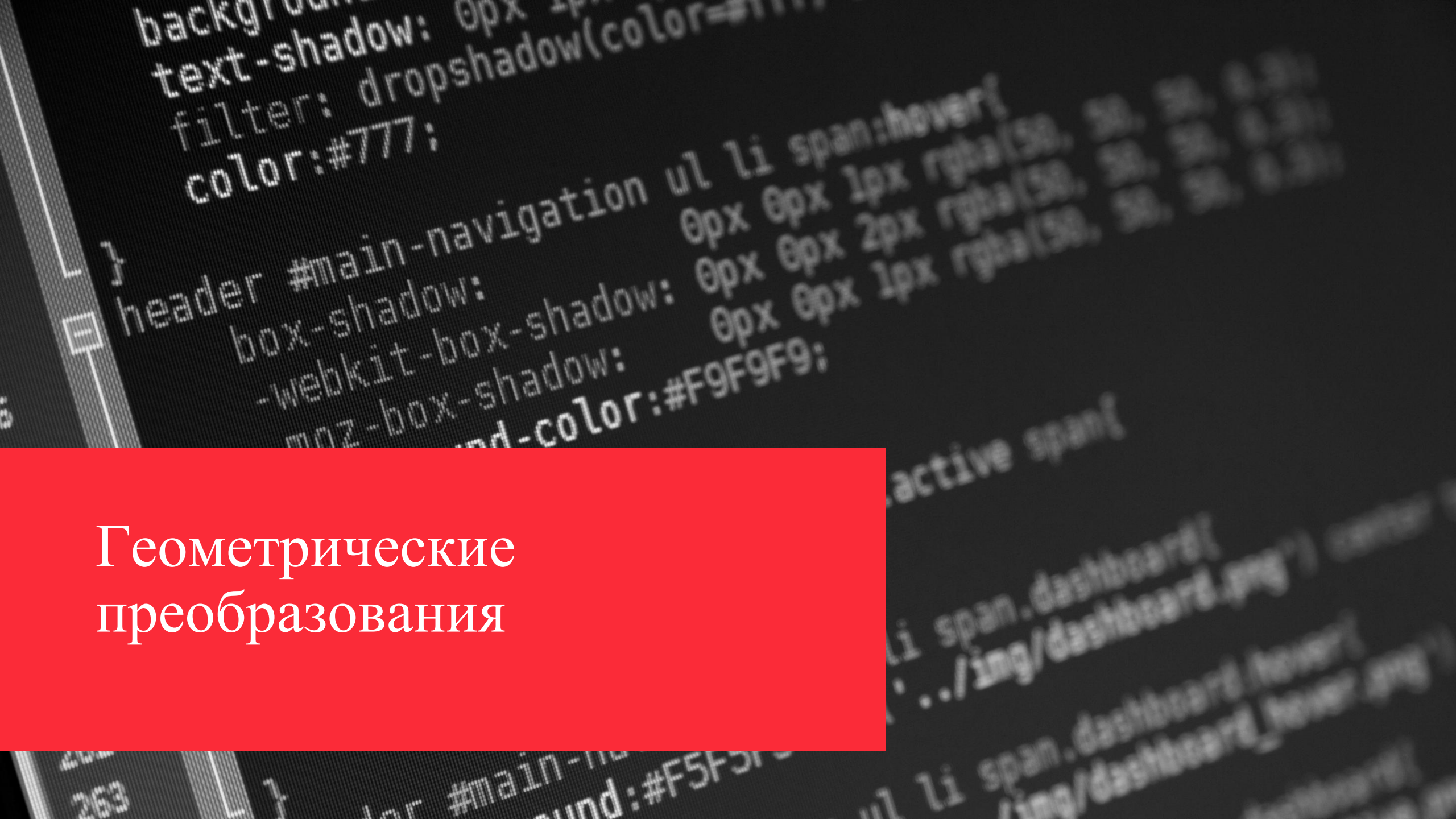




Опрос

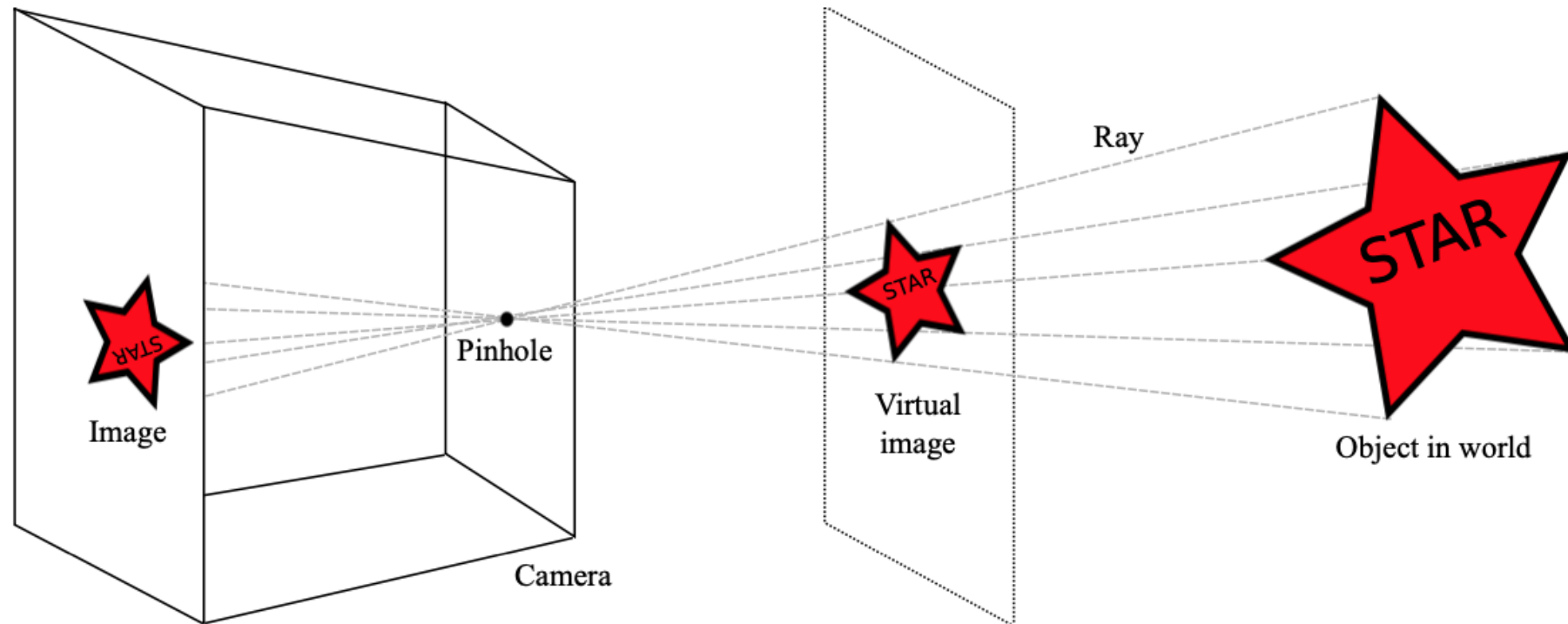
<https://forms.gle/E97DHTgfapd1sh4P7>

Геометрические преобразования

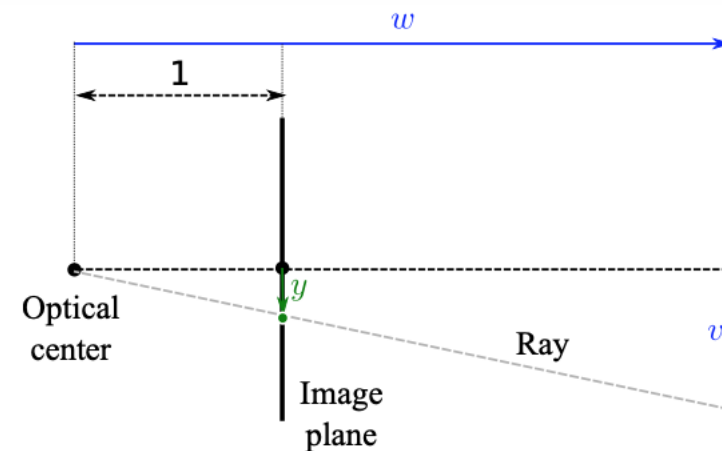
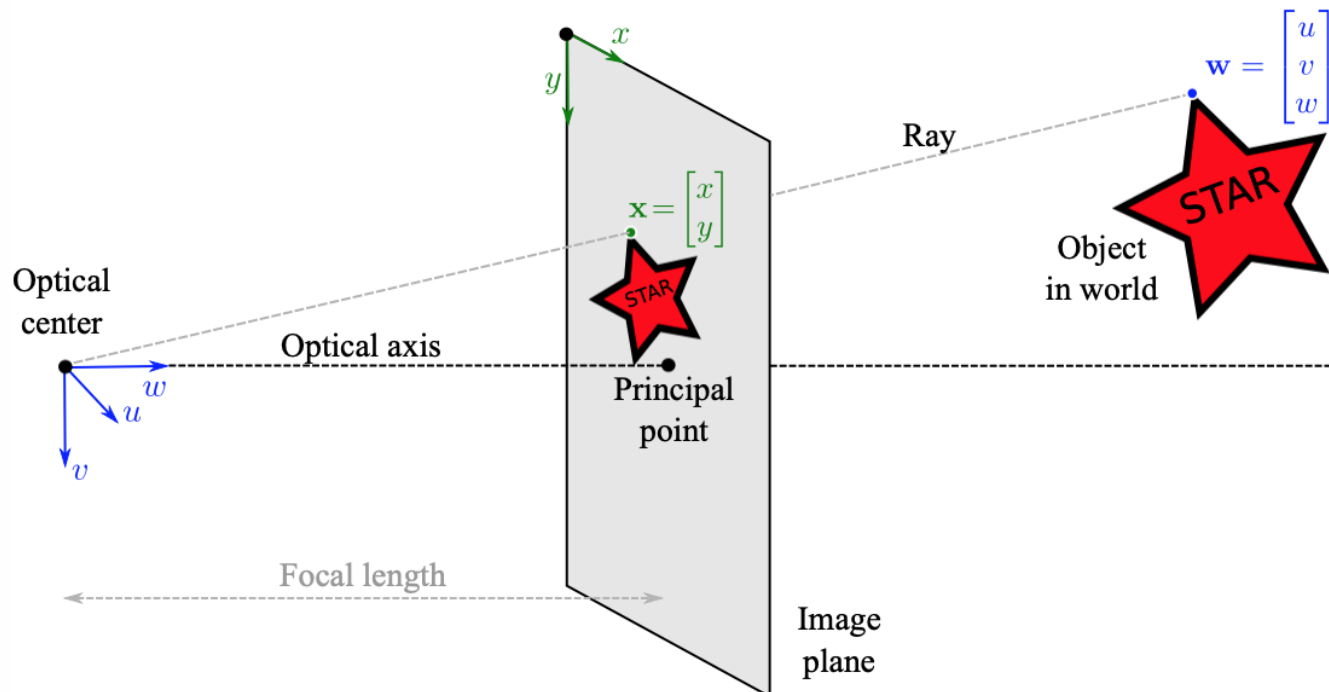


Простейшая модель оптической системы

Pinhole camera (камера-обскура)



Геометрическая модель

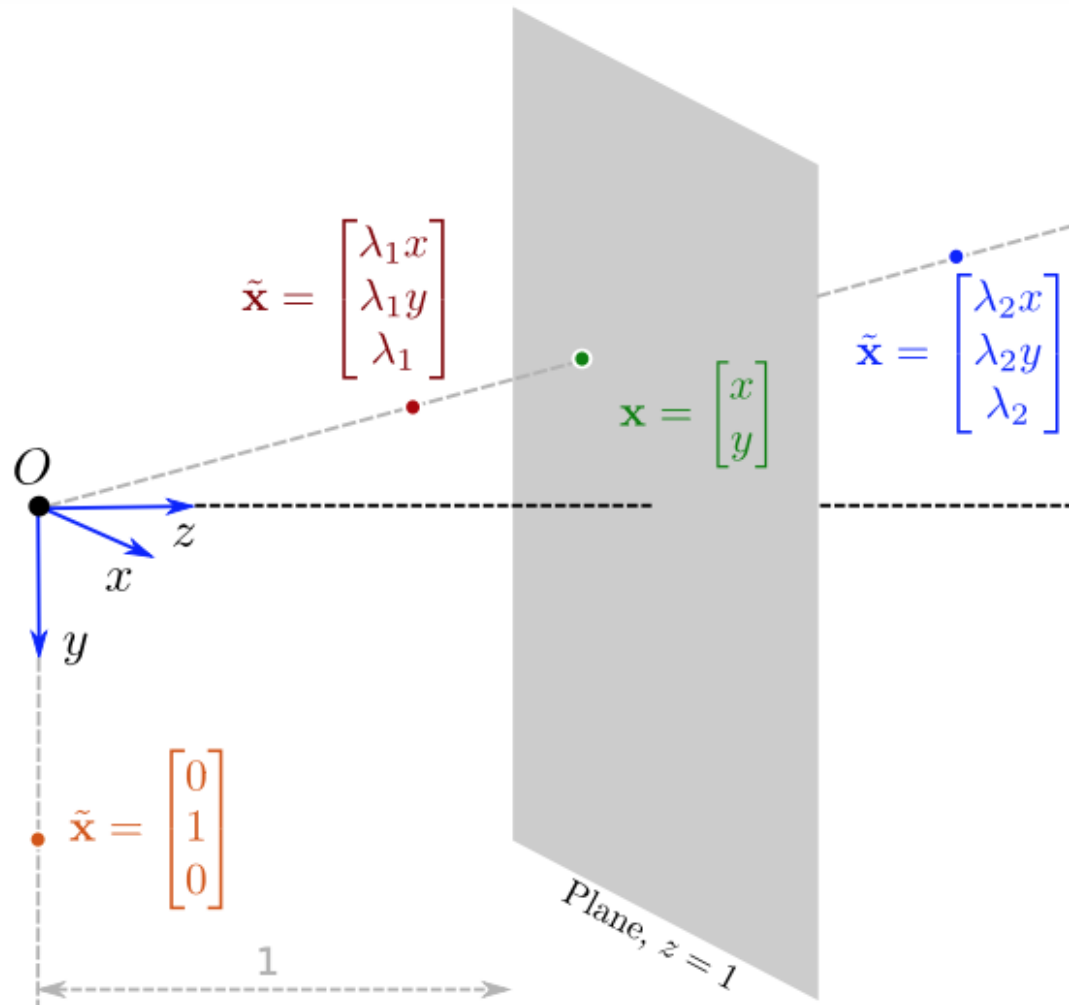


Упрощенная модель (камера находится в начале глобальной системы координат)

$$x = \frac{\phi_x u + \gamma v}{w} + \delta_x$$

$$y = \frac{\phi_y v}{w} + \delta_y.$$

Однородные (homogeneous) координаты



$$\tilde{\mathbf{x}} = \lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



2D

$$\begin{aligned} x &= \frac{\tilde{x}}{\tilde{z}} \\ y &= \frac{\tilde{y}}{\tilde{z}} \end{aligned}$$

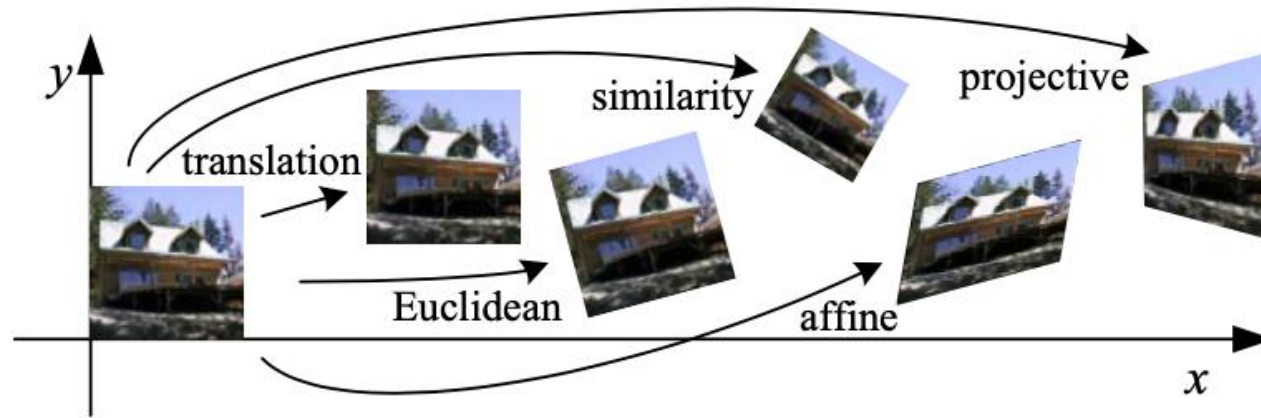
3D

$$\tilde{\mathbf{w}} = \lambda \begin{bmatrix} u \\ v \\ w \\ 1 \end{bmatrix}$$

Модель камеры-обскуры

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \phi_x & \gamma & \delta_x & 0 \\ 0 & \phi_y & \delta_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ 1 \end{bmatrix}$$

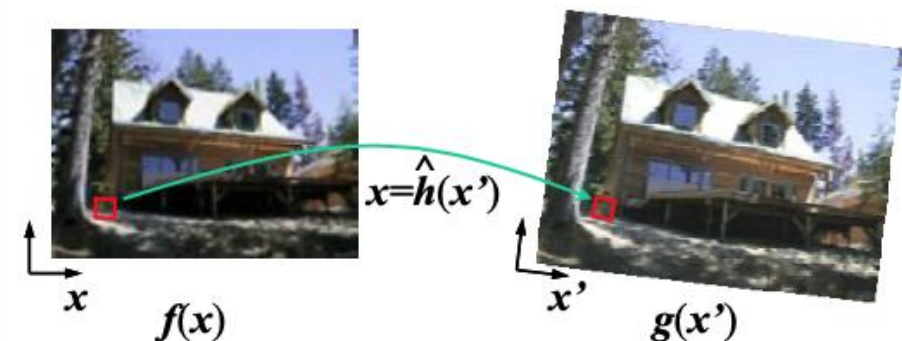
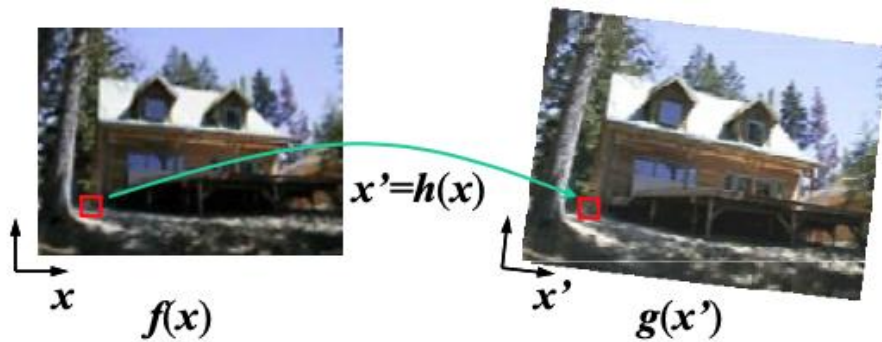
Основные преобразования



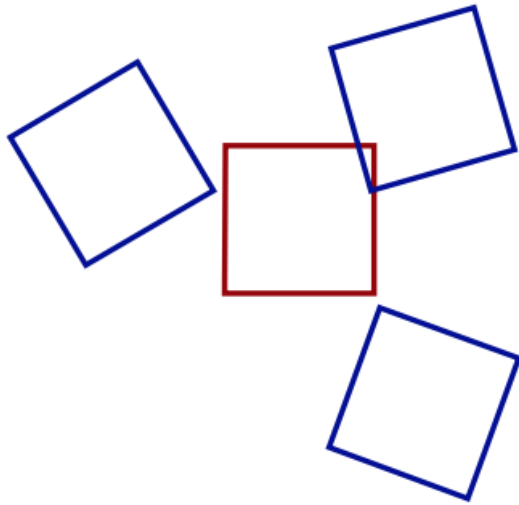
$$g(x) = f(h(x))$$

Forward warping

Inverse warping



Преобразование Евклида: перенос и поворот



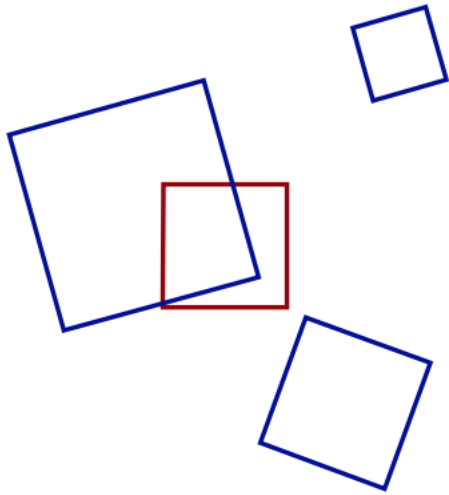
$$\mathbf{x}' = \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \bar{\mathbf{x}}$$

$$\mathbf{R} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

Однородные координаты

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \omega_{11} & \omega_{12} & \tau_x \\ \omega_{21} & \omega_{22} & \tau_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$
$$\begin{bmatrix} \omega_{11} & \omega_{12} \\ \omega_{21} & \omega_{22} \end{bmatrix} = \begin{bmatrix} \cos[\theta] & \sin[\theta] \\ -\sin[\theta] & \cos[\theta] \end{bmatrix}$$

Преобразование подобия (similarity transform)



$$\mathbf{x}' = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \end{bmatrix} \bar{\mathbf{x}} = \begin{bmatrix} a & -b & t_x \\ b & a & t_y \end{bmatrix} \bar{\mathbf{x}},$$

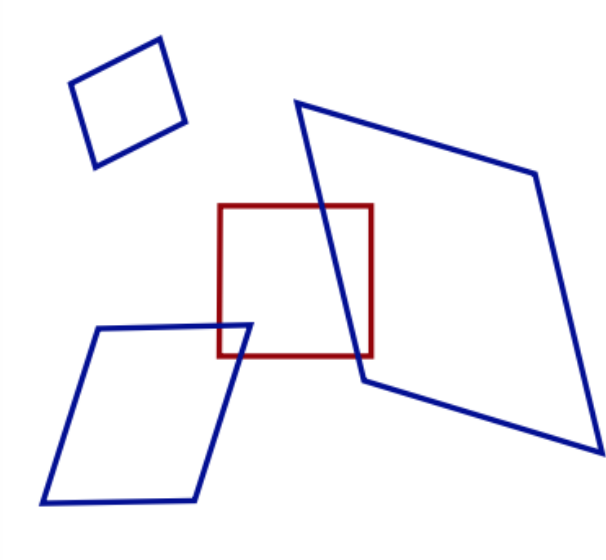
Однородные координаты

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \rho\omega_{11} & \rho\omega_{12} & \tau_x \\ \rho\omega_{21} & \rho\omega_{22} & \tau_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} \omega_{11} & \omega_{12} \\ \omega_{21} & \omega_{22} \end{bmatrix} = \begin{bmatrix} \cos[\theta] & \sin[\theta] \\ -\sin[\theta] & \cos[\theta] \end{bmatrix}.$$

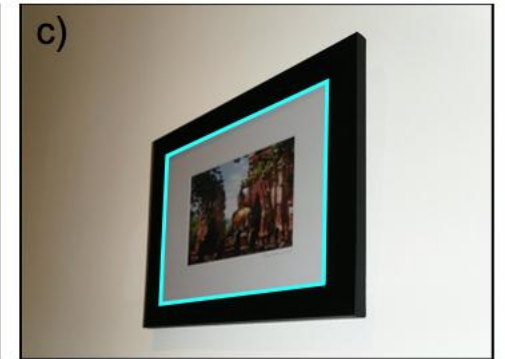
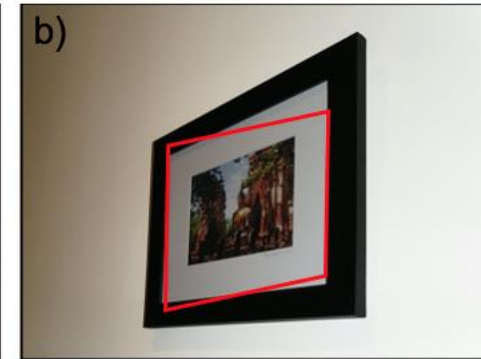
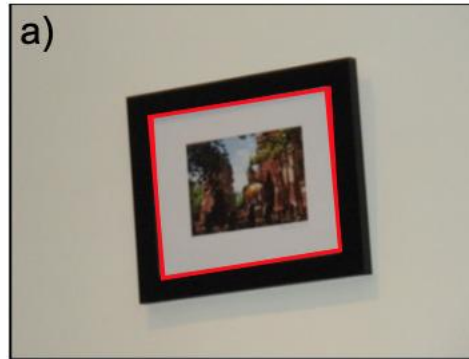
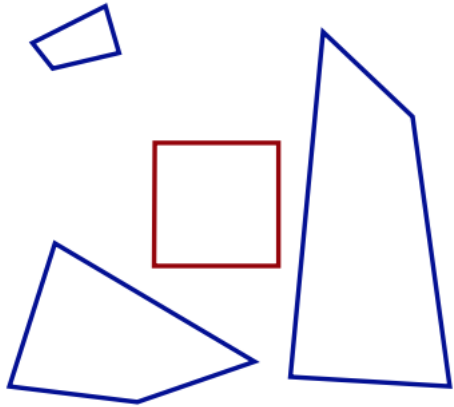
Аффинное преобразование

Параллельные линии остаются параллельными



$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \phi_{11} & \phi_{12} \\ \phi_{21} & \phi_{22} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} \tau_x \\ \tau_y \end{bmatrix}$$

Проективное (перспективное) преобразование



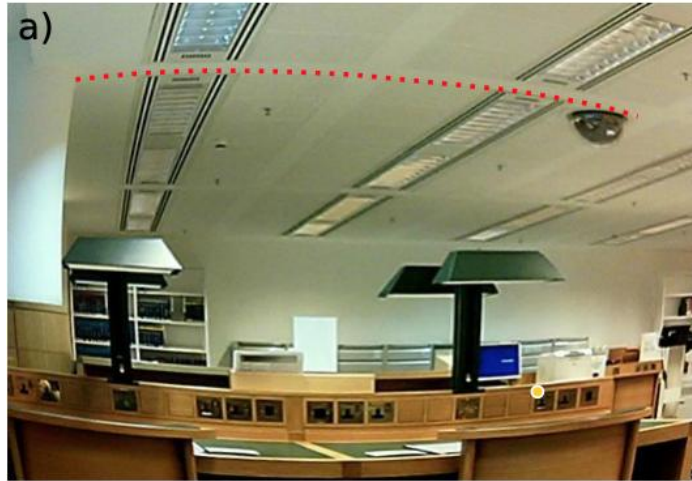
Однородные координаты

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} \\ \phi_{21} & \phi_{22} & \phi_{23} \\ \phi_{31} & \phi_{32} & \phi_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$



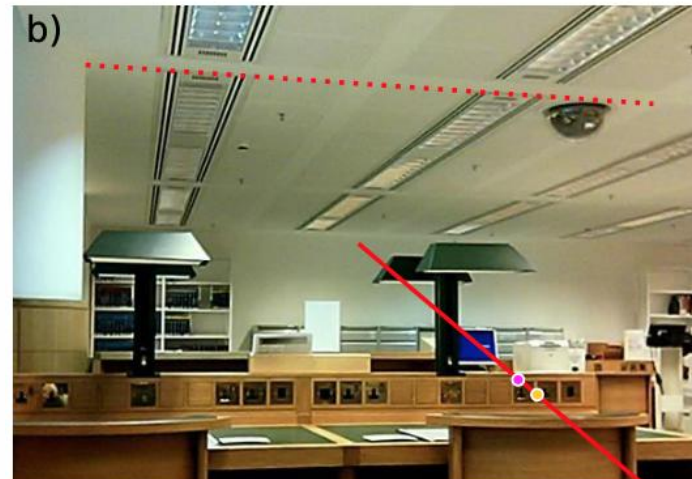
$$\begin{aligned} x &= \frac{\phi_{11}u + \phi_{12}v + \phi_{13}}{\phi_{31}u + \phi_{32}v + \phi_{33}} \\ y &= \frac{\phi_{21}u + \phi_{22}v + \phi_{23}}{\phi_{31}u + \phi_{32}v + \phi_{33}}, \end{aligned}$$

Искажения перспективной проекции. Радиальная дисторсия



inverse radial
distortion model


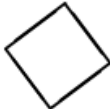



$$\begin{aligned}x' &= x(1 + \beta_1 r^2 + \beta_2 r^4) \\ y' &= y(1 + \beta_1 r^2 + \beta_2 r^4).\end{aligned}$$



“Fisheye”



Геометрические преобразования (итог)

Transformation	Matrix	# DoF	Preserves	Icon
translation	$\begin{bmatrix} \mathbf{I} & \mathbf{t} \end{bmatrix}_{2 \times 3}$	2	orientation	
rigid (Euclidean)	$\begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}_{2 \times 3}$	3	lengths	
similarity	$\begin{bmatrix} s\mathbf{R} & \mathbf{t} \end{bmatrix}_{2 \times 3}$	4	angles	
affine	$\begin{bmatrix} \mathbf{A} \end{bmatrix}_{2 \times 3}$	6	parallelism	
projective	$\begin{bmatrix} \tilde{\mathbf{H}} \end{bmatrix}_{3 \times 3}$	8	straight lines	

#DoF – число степеней свободы (параметров)

Нелинейная фильтрация

Медианный фильтр (1)

$\{ 5, 7, 3, 4, 5, 19, 6, 4, 9 \} \xrightarrow{\text{sort}} \{ 3, 4, 4, 5, \boxed{5}, 6, 7, 9, 19 \}$

123	125	126	130	140
122	124	126	127	135
118	120	150	125	134
119	115	119	123	133
111	116	110	120	130

Neighbourhood values:

115, 119, 120, 123, 124,
125, 126, 127, 150

Median value: 124

Медиана нелинейна:

$\text{med}(\{1,2,3\} + \{5,4,6\}) = \text{med}(\{6,6,9\}) = 6$

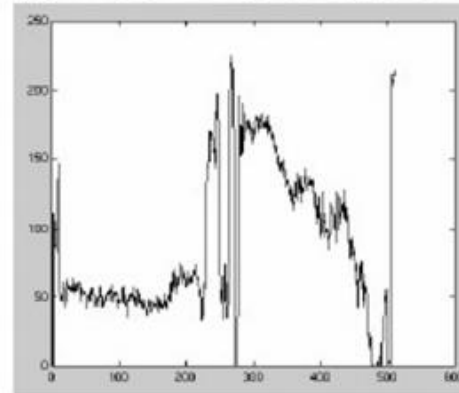
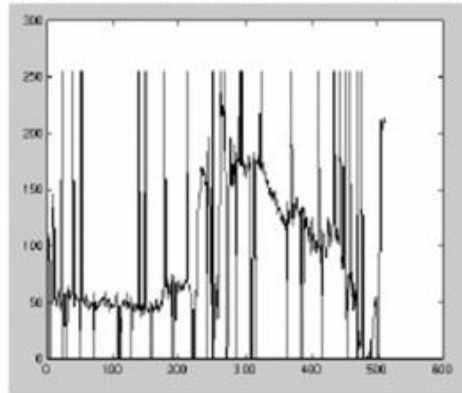
$\text{med}(\{1,2,3\}) + \text{med}(\{5,4,6\}) = 2 + 5$

Медианный фильтр (2). Шум «Salt-and-pepper»

**Salt and
pepper
noise**



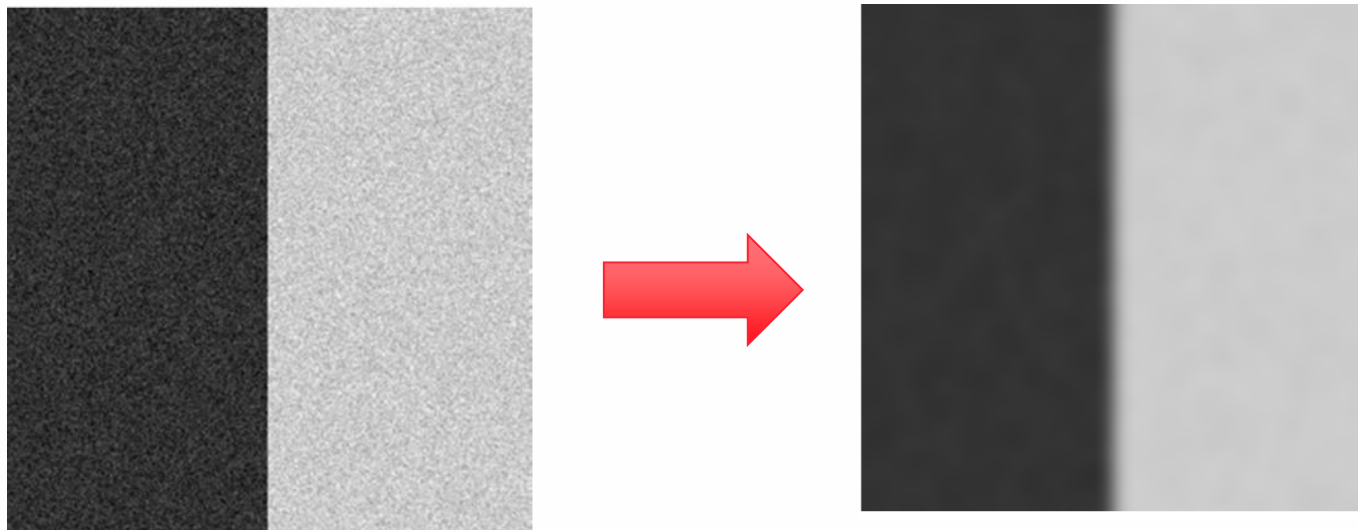
**Median
filtered**



Plots of a row of the image

Билатеральный фильтр (1)

Фильтр Гаусса сглаживает не только шум, но и границы объектов



$$w(i, j, k, l) = \exp \left(-\frac{(i - k)^2 + (j - l)^2}{2\sigma_d^2} - \frac{\|f(i, j) - f(k, l)\|^2}{2\sigma_r^2} \right)$$

Domain kernel Range kernel

$$g(i, j) = \frac{\sum_{k, l} f(k, l) w(i, j, k, l)}{\sum_{k, l} w(i, j, k, l)}$$

OpenCV: `bilateralFilter()`

Билатеральный фильтр (2)

Noisy input



Bilateral filter 7x7 window



Математическая морфология (1)

Структурные элементы

1	1	1
1	1	1
1	1	1

	1	
1	1	1
	1	

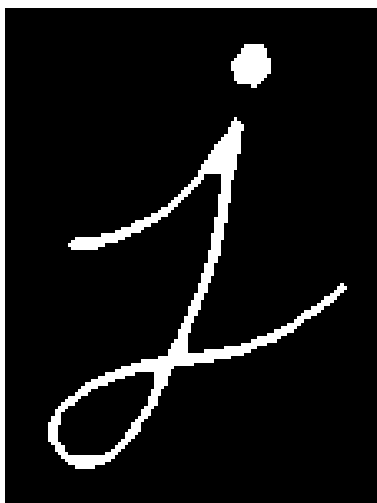
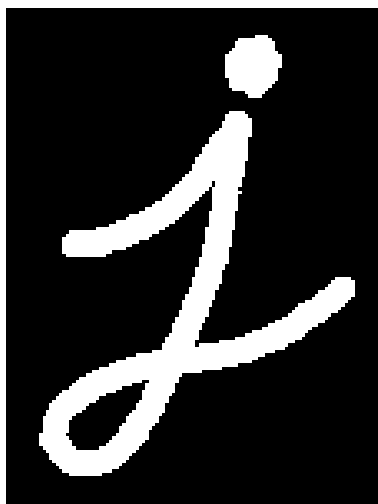
		1	1	1		
	1	1	1	1	1	
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
	1	1	1	1	1	
		1	1	1		

1	1	
1	0	
1		0

1	1	1
1	10	1
1	1	1

Математическая морфология (2)

Сужение (Erosion)



Пиксель=1, только если все пиксели, соответствующие структурному элементу, равны 1

Расширение (Dilation)



Пиксель=1, если хотя бы один пиксель, соответствующий структурному элементу, равен 1

OpenCV:

- `erode()`,
- `dilate()`

Структурный элемент – квадрат 5x5

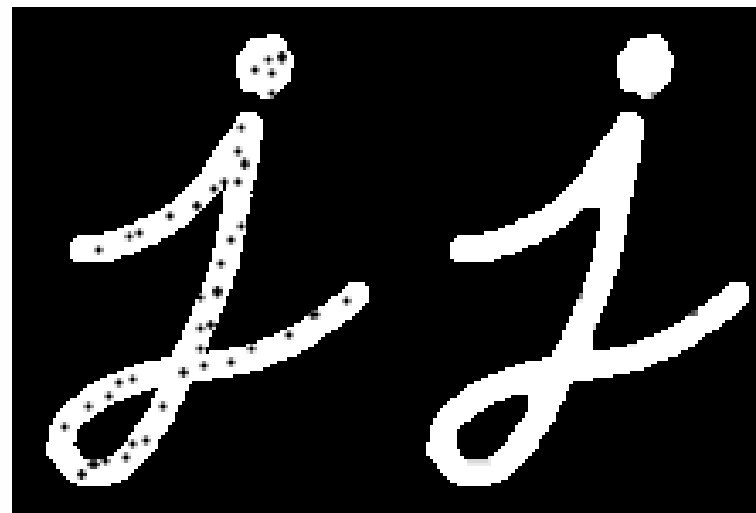
Математическая морфология (3)

Открытие (Opening)



erosion followed by dilation –
очистка фона от шума

Заккрытие (Closing)



Dilation followed by Erosion –
закрываются «дыры» на объекте

OpenCV: `morphologyEx()` //MORPH_OPEN, MORPH_CLOSE

Параметр `iterations` – число сужений и расширений

Ex. 2 для `morph_open`: `erode -> erode -> dilate -> dilate`

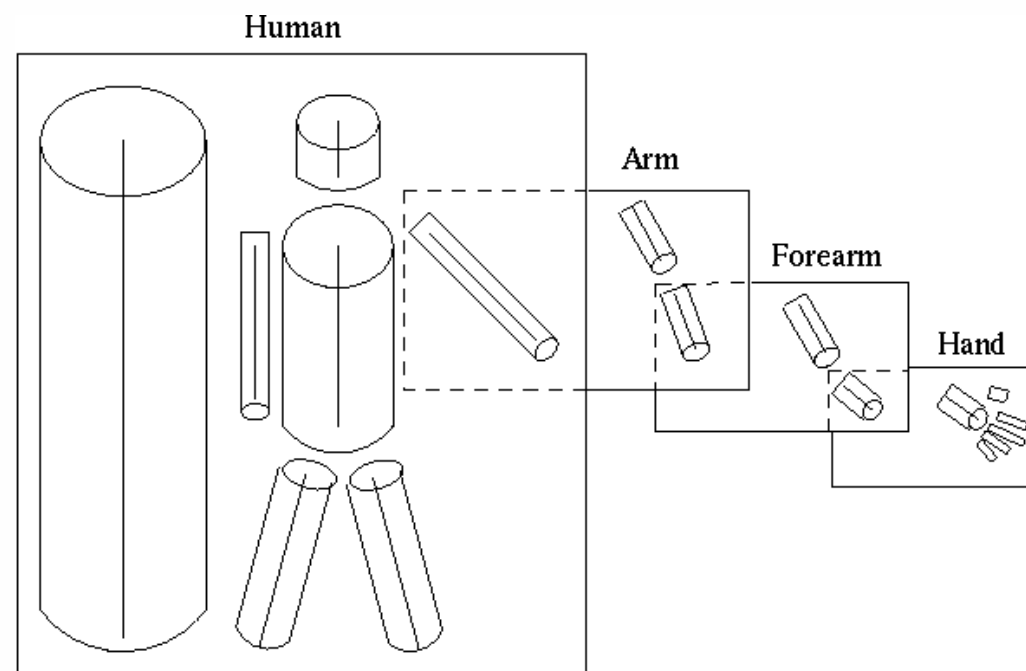
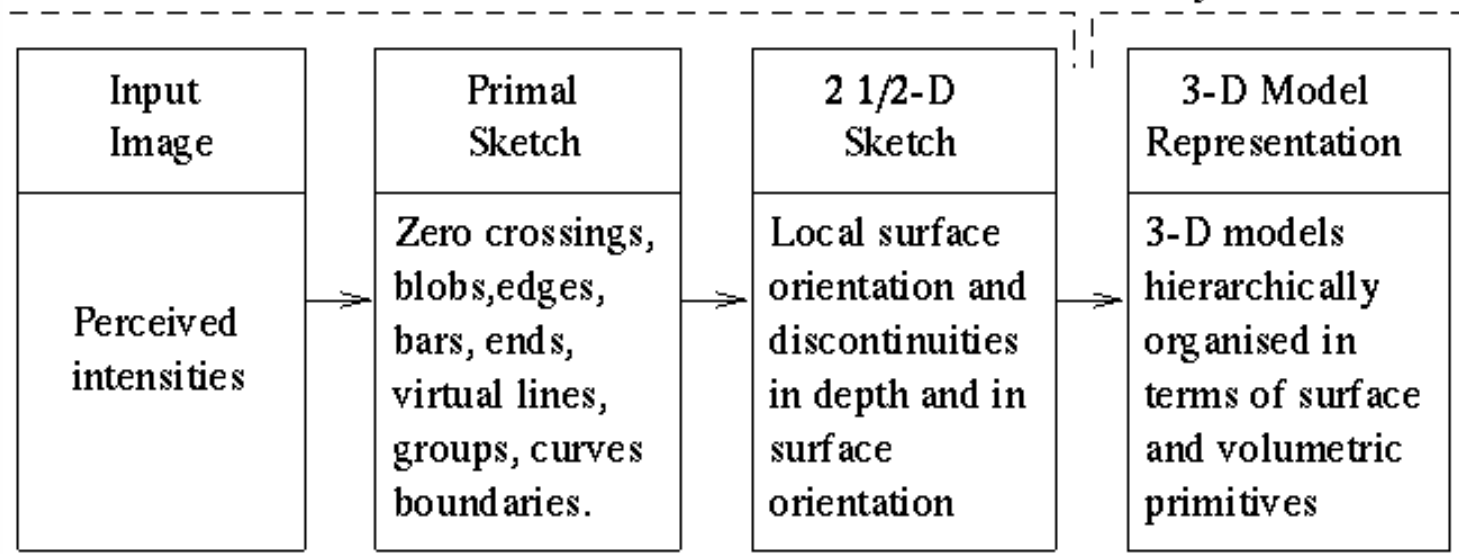
Структурный элемент – квадрат 5x5

Выделение краев (edge detection)

Информационная теория зрительного восприятия (David Marr)

Viewer centred

Object centred

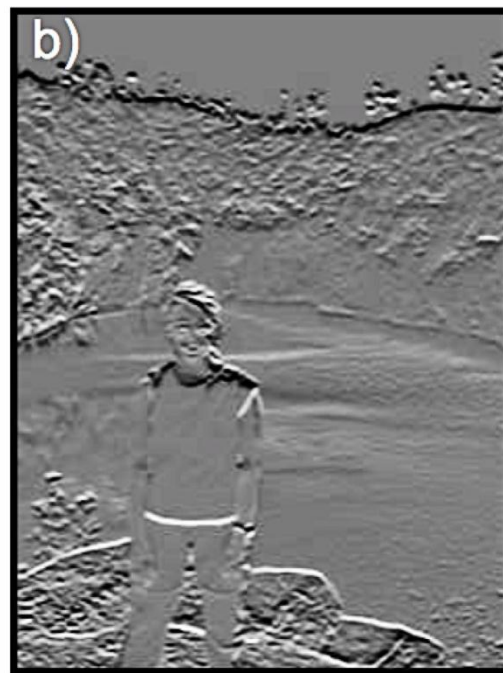


Вычисление первых производных (1). Фильтры Превитт

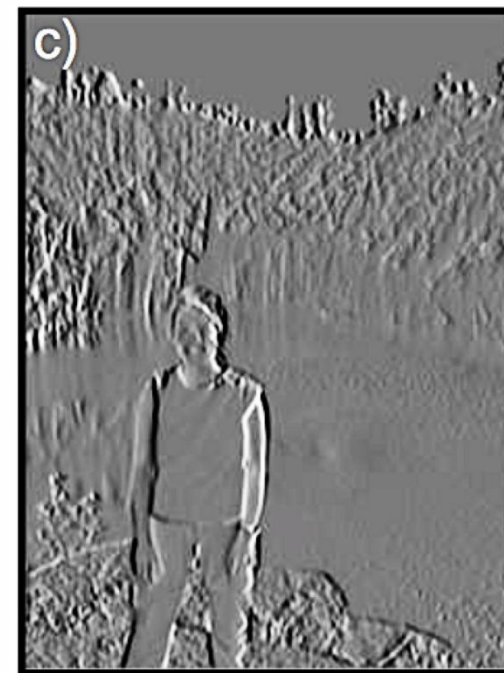
$$\mathbf{F}_x = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}, \quad \mathbf{F}_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$



Original image



Prewitt (vertical)



Prewitt (horizontal)

Вычисление первых производных (2). Фильтры Собеля

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * I \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * I$$

Модуль градиента

$$G = \sqrt{G_x^2 + G_y^2} \quad G = |G_x| + |G_y|$$

Оператор Шарра

$$G_x = \begin{bmatrix} -3 & 0 & +3 \\ -10 & 0 & +10 \\ -3 & 0 & +3 \end{bmatrix}$$

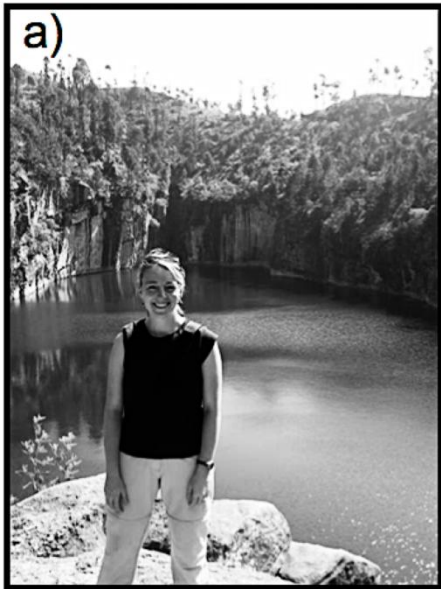
$$G_y = \begin{bmatrix} -3 & -10 & -3 \\ 0 & 0 & 0 \\ +3 & +10 & +3 \end{bmatrix}$$



OpenCV:

- Sobel(),
- Sharr()

Вторые производные. Фильтр Лапласа.



Original image

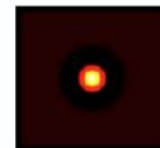


Laplacian

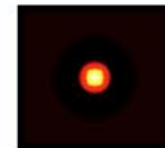
$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$



Laplacian of Gaussian



Difference of Gaussians



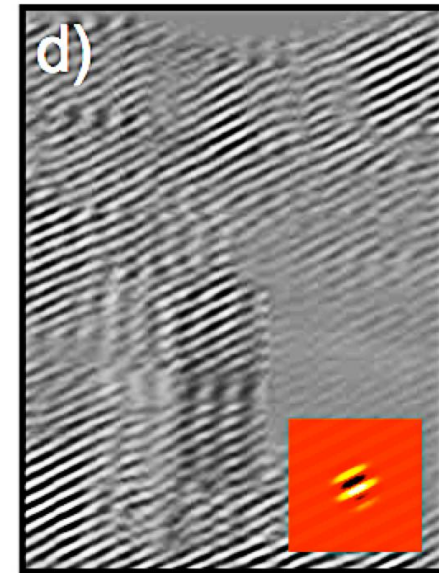
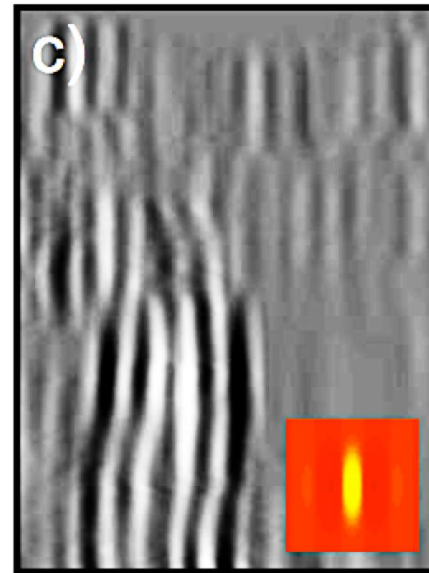
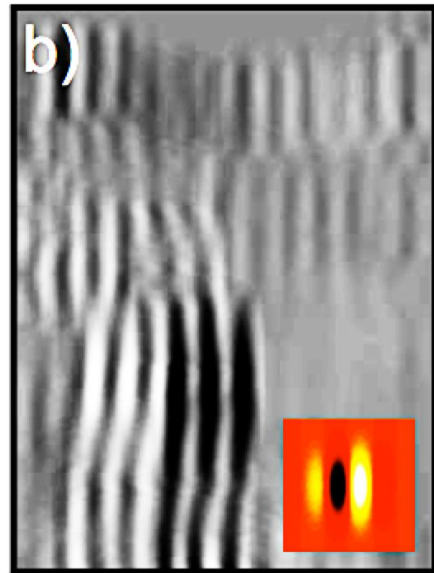
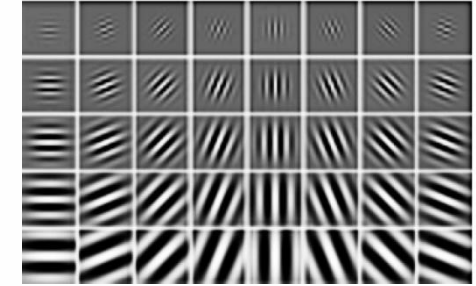
OpenCV:

- `Laplacian()`
- `filter2D`

$$LoG(x; \sigma) = \left(\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2} \right) G(x; \sigma)$$

Фильтры Габора

$$f_{mn} = \frac{1}{2\pi\sigma^2} \exp \left[-\frac{m^2 + n^2}{2\sigma^2} \right] \sin \left[\frac{2\pi(\cos[\omega]m + \sin[\omega]n)}{\lambda} + \phi \right]$$



Оператор Кэнни

b) Вертикальный фильтр Превитта

c) Горизонтальный фильтр Превитта

d) Квантование ориентации

$$\theta_{ij} = \arctan[v_{ij}/h_{ij}]$$

e) Амплитуда градиента

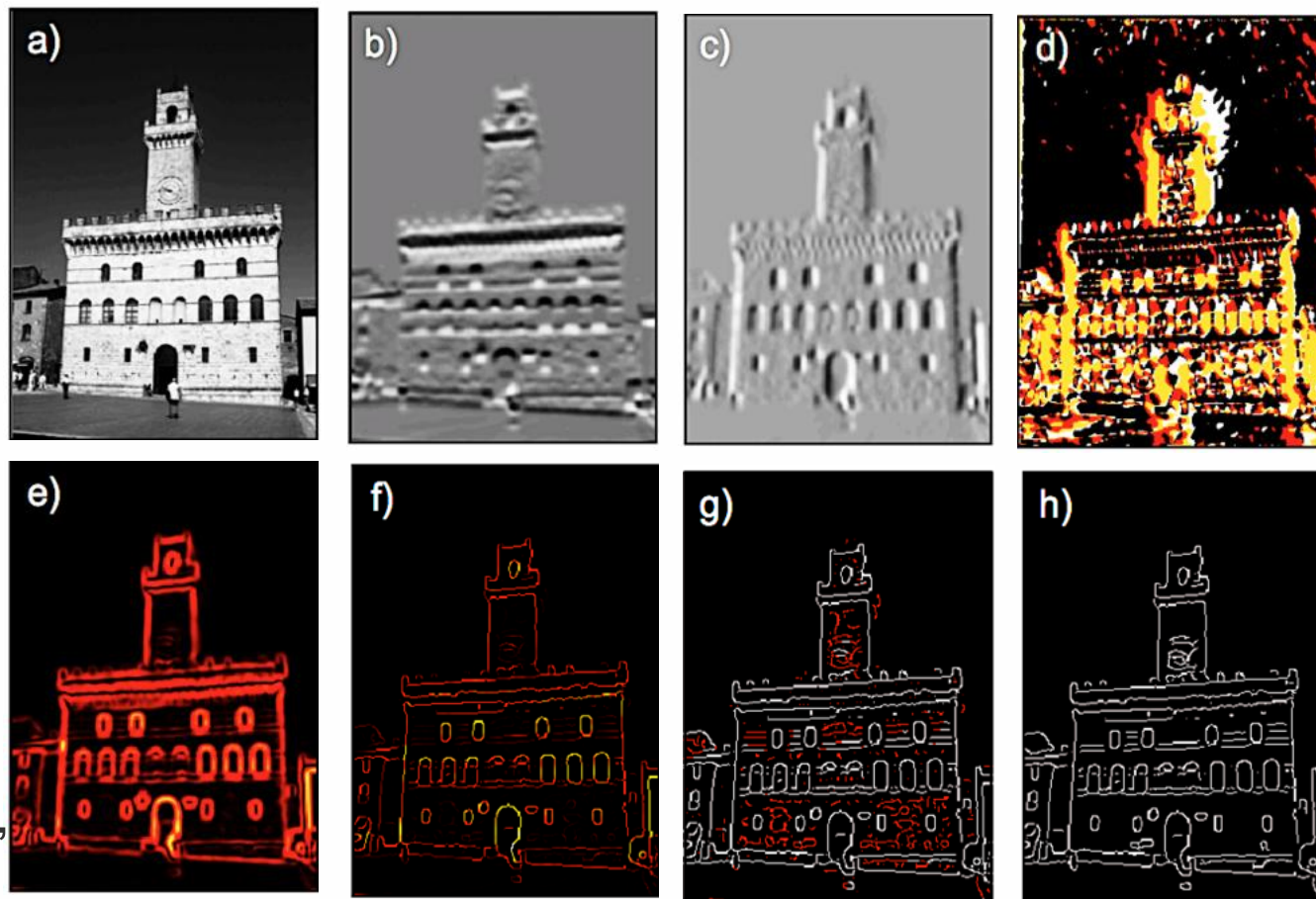
$$a_{ij} = \sqrt{h_{ij}^2 + v_{ij}^2}$$

f) Non-maximal suppression: угол квантуется ($\{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$), амплитуда:=0, если меньше хотя бы 1 из 2 перпендикулярных градиенту

g) Два порога:

белые пиксели выше максимального порога, красные – больше минимального

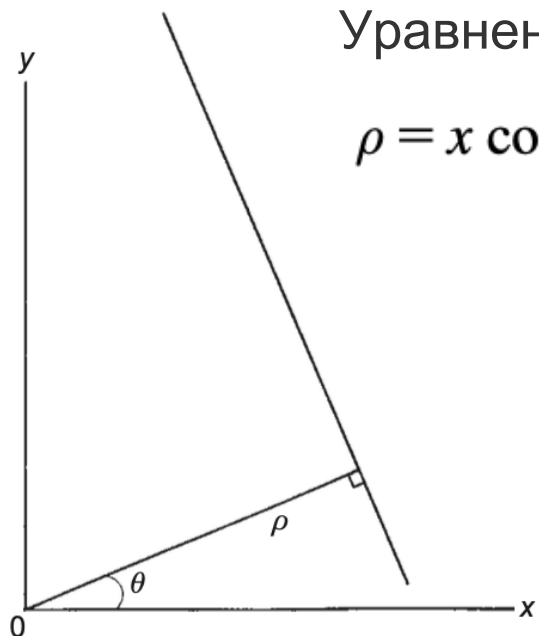
h) Hysteresis thresholding – все белые пиксели и связанные с ними красные



OpenCV: `canny()`

Преобразование Хафа

Поиск прямых



Уравнение прямой

$$\rho = x \cos \theta + y \sin \theta$$

Для каждой точки

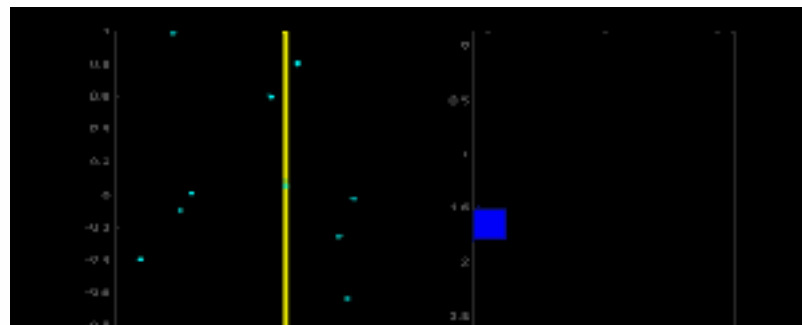
1) найдем направление градиента

$$\theta = \arctan\left(\frac{g_y}{g_x}\right)$$

2) оценим ρ и добавим 1 в аккумуляторный массив (ρ, θ)

В аккумуляторе квантуются значения угла θ и расстояния (от 0 до длины диагонали)

Выберем пары (ρ, θ) , для которых счетчик превышает порог



Progressive Probabilistic Hough Transform: выбирается часть (до 20%) точек

OpenCV:

- `HoughLines()`
- `HoughLinesP()`

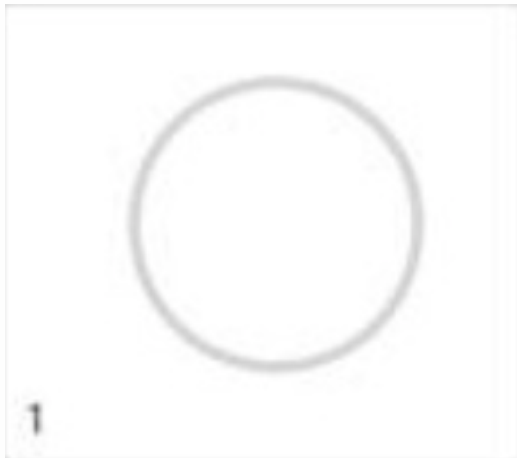
Davies «Computer vision»

https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_houghlines/py_houghlines.html
<https://core.ac.uk/download/pdf/47168821.pdf>

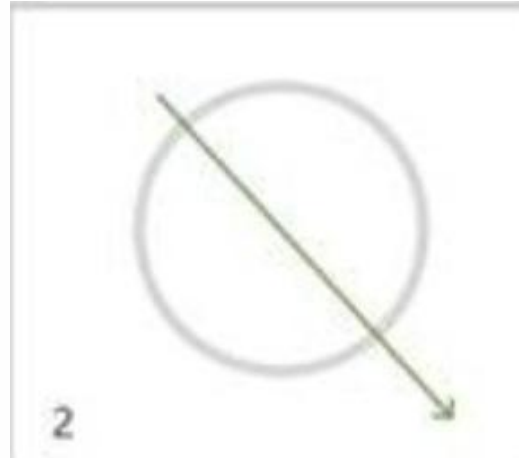
Поиск окружностей. Hough gradient method

Уравнение окружности
 $(x-x_c)^2+(y-y_c)^2=R^2$

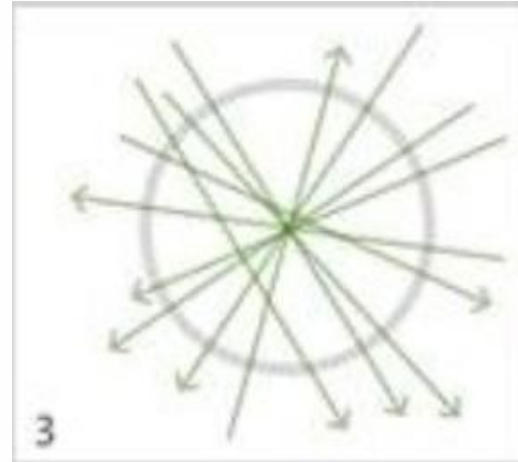
Три параметра – нужен
трехмерный аккумулятор



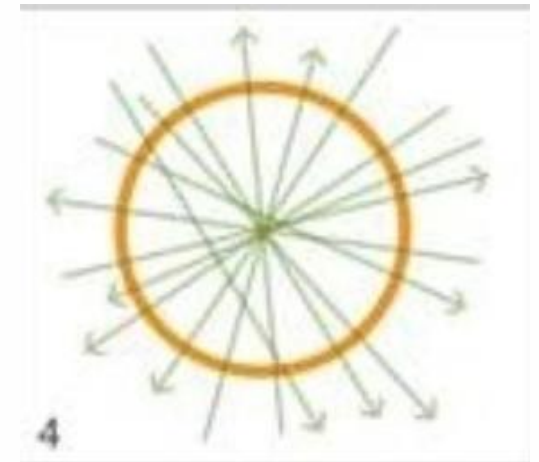
Edge detector
(Canny)



Увеличивается
аккумулятор для θ



Находятся центры
– голосование



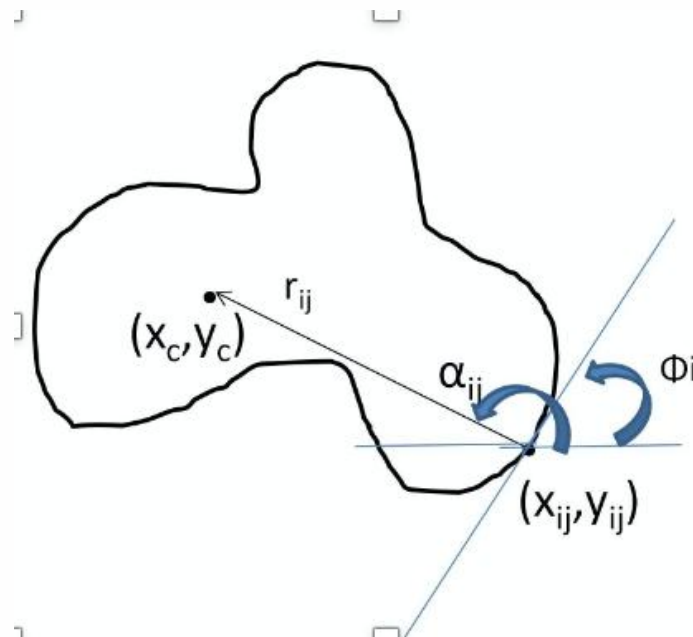
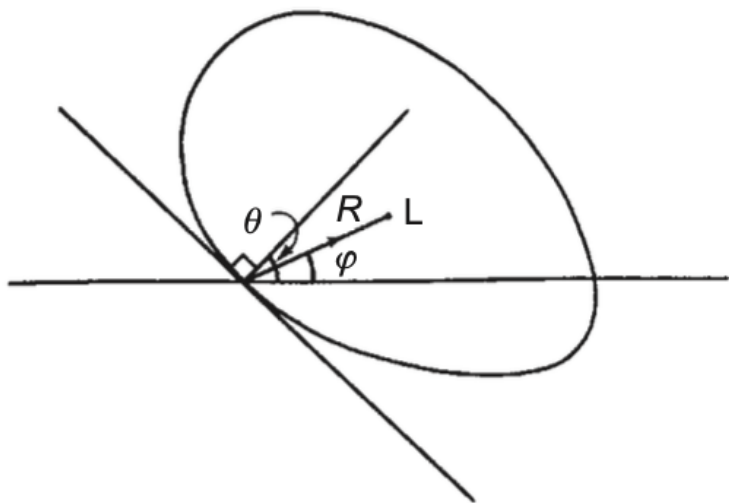
Для центра определяется радиус
(максимум точек контура)

OpenCV: `HoughCircles()`

Обобщенное преобразование Хафа

Произвольные кривые

Нужна опорная точка и функции (от нормали) расстояния $R(\theta)$ и угла $\varphi(\theta)$ до нее от любой граничной точки



Центр ищется голосованием на основе всех точек краев



Перейдем к примерам

<https://github.com/HSE-asavchenko/MADE-mobile-image-processing/tree/master/lesson3/src>